IN THE SPECIFICATION

Please amend the Title on page 1, line 2 as follows:

A Data Processor That Suspends A Cache Fill Operation Due to an Interrupt Request or Branch Instruction ~~Having Cache Memory~~

Please replace the paragraph at page 1, lines 17-25, with the following rewritten paragraph:

Conventional data processors suffer from the following problems. When an interrupt request occurs during a burst transfer, an interrupt processing is started after the completion of the burst transfer. Therefore, a response to the interrupt request is late. Likewise, when a branch instruction to ~~other~~ another program is detected during a burst transfer, an instruction to fetch [[of]] a branch target program is started after the completion of the burst transfer. Therefore, the execution of the branch target program is late. Further, if not returning to the original program at the completion of the interrupt processing, or at the completion of the execution of the branch target program, unnecessary data ~~exist~~ exists in the cache memory.

Please replace the paragraph at page 2, lines 14-18, with the following rewritten paragraph:

According to a second aspect of the present invention, a data processor ~~processing~~ processes instructions in a pipeline having an instruction fetch stage ~~fetching~~ that fetchesthe instructions from a memory, a decode stage ~~decoding~~ that decodes the instructions fetched by the fetch stage, and an instruction execution stage ~~executing~~ that executes the instructions decoded by the decode stage.

Please replace the paragraph at page 3, lines 5-10, with the following rewritten paragraph:

An interrupt which needs an urgent interrupt processing can be started immediately by suspending the instruction that is already fetched. On the other hand, if the priority of an interrupt request is not so high, an interrupt handler is executed after executing the instruction that is already fetched before the interrupt. Therefore, the processing proceeds without disturbing the pipeline, and the penalty caused by the interrupt processing can be eliminated.

Please replace the paragraph at page 7, lines 3-11, with the following rewritten paragraph:

~~Following is~~ The following are signals given and received between the CPU 1 and operand cache 2. An operand fetch request signal OFR is a signal indicting that the CPU 1 requests an operand fetch from the operand cache 2. An operand address signal OFA is a signal indicating the address of an operand required by the CPU 1. A receipt signal AFOC is a signal indicating that the operand cache 2 accepts the operand fetch request from the CPU 1. A read operand RO is an operand that is transferred from the operand cache 2 to the CPU 1. An operand valid signal OV is a signal indicating that a valid operand is transferred from the operand cache 2 to the CPU 1. A write operand WO is an operand that is transferred from the CPU 1 to the operand cache 2. Each of the operand address signal OFA, the read operand RO and the write operand WO is a 32-bit bus signal.

Please replace the paragraph at page 7, line 18 to page 8, line 2 with the following rewritten paragraph:

Fig. 2 is a block diagram showing the instruction cache 3 and SDRAM controller 5. ~~Following is~~ The following are signals given and received between the instruction cache 3

and SDRAM controller 5. An address signal IA is a signal that indicates the address of an instruction code required by the CPU 1 and that is inputted from the instruction cache 3 to the SDRAM controller 5 via an address bus AB shown in Fig. 1. A bit ~~wide~~ width of the address bus AB is 32 bits in this example. An access request signal AR is a signal indicating that the instruction cache 3 requests access to the SDRAM 8. A burst request signal BR is a signal requiring that data be transferred from the SDRAM 8 to the instruction cache 3 by burst transfer. When the burst request signal BR is "H (high)", burst transfer is designated, and when it is "L (low)", one-shot transfer is designated.

Please replace the paragraph at page 8, lines 3-12 with the following rewritten paragraph:

A last request signal LAR is valid when a burst transfer is designated, and is a signal indicating the last access request in the burst transfer. An acknowledge signal ACK is a signal indicating that the SDRAM controller 5 accepts an access request from the instruction cache 3. A data ready signal DR is a signal indicating that a valid data is transferred from the SDRAM controller 5 to the instruction cache 3. A read data RD is data transferred from the SDRAM controller 5 to the instruction cache 3 via a data bus RDB shown in Fig. 1. A bit ~~wide~~ width of the data bus RDB is 32 bits in this example. The data bus WDB transfers data from the operand cache 2 and the instruction cache 3 to each of memories 8, 9 and 10 for writing to each of memories 8, 9 and 10. A bit ~~wide~~ width of the data bus WDB is 32 bits in this example. The read data RD is a 32-bit bus signal.

Please replace the paragraph at page 8, lines 13-20 with the following rewritten paragraph:

The SDRAM controller 5 inputs an address signal IA and control signals CS, RAS, CAS, and WE to the SDRAM 8 via an address control bus ACB shown in Fig. 1. In this example, a bit ~~wide~~ width of the address control bus ACB is 36 bits in total, including 32 bits for transferring the address signal IA and 4 bits for transferring the control signals CS, RAS, CAS and WE. Further, data DQ are given and received between the SDRAM controller 5 and SDRAM 8, via a data bus DB shown in Fig. 1. A bit ~~wide~~ width of the data bus DB is 32 bits in this example. Each of the address signal IA and the data DQ is a 32-bit bus signal.

Please delete the Abstract on page 48, lines 1-10, and replace it with the following new Abstract: